COMP1204: Data Management
Coursework One: Hurricane Monitoring

Orasiki Wellington
oew1g21

March 15, 2022

1 Introduction

The aim of this report is to analyse a bash script that data cleans hurricane monitoring data. When given a KML (Keyhole Markup Language) file, this script will generate a CSV (comma-separated values) file which contains relevant information from the KML file, and is used to create storm plots. This is seen in practise when 3 different storm plots are created from 3 different KML files with the help of the script.

This report consists of two main sections. One explains the inner workings of the script, while the other shows the storm plots generated from the KML files.

2 Create CSV Script

2.1 Script

Listing 1: Bash Script Code

```
#!/bin/bash
awk -F "</*td>|</*tr>" '/<td>.*/ {print $3}' "$1" | sed -e 's/\(.<B>\|<\/B
    >.\|;.*\)//g' -e '/hr/d' -e's/\(, \)/,/g'| awk '(NR%2==0) {print}' |
    awk '{line=line "," $0}  (NR%4==0) {print substr(line,2); line=""}'|
    sed -e '1 i\Timestamp,Latitude,Longitude,MinSeaLevelPressure,
    MaxIntensity' > "$2"
```

2.2 Overall Explanation

In total, the bash script consists of 6 programs joined together using pipes. These programs will be explained in detail, starting with the first line of the script.

Listing 2: First line of Bash script

```
#!/bin/bash
```

The first line of the bash script indicates to the system's operating system to call on the bash shell to run the commands that come after it in the script.

Listing 3: First program

```
#awk program gets contents between td and tr tags
awk -F "</*td>|</*tr>" '/<td>.*/ {print $3}' "$1"
```

The first program retrieves the kml file passed as the first argument of the script. It uses the awk command with the -F option to tell awk that the field separators are the closing and opening td or tr tags in the file. This is easily done using wildcards. Next, the program fetches all records containing td tags and prints the third field. This returns the contents between the td tags. The output of this program is piped to the second program which is a sed program.

Listing 4: Second program

```
#sed program filters out tags and data we don't need
sed -e 's/\(.<B>\|<\/B>.\|;.*\)//g' -e '/hr/d' -e's/\(, \)/,/g'
```

The second program is a sed program that works on the output of the first program. The sed program is used with the -e option which tells sed to execute the command line arguments after this as a sed program.

The sed program has many functions. One function of the sed program is to remove all lines containing the hr tag. Another function of the sed program is to remove the B opening and closing tags and everything that comes after a semi colon in all lines of the script. Lastly, the sed program removes all spaces after a comma. The last two deletions the sed program performs are inline and are done via global substitution with the help of wildcards. This differs from the first case mentioned which deletes the whole line rather than words in a line. The output of the sed program is piped to the third program; another awk program.

Listing 5: Third program

```
#awk program skips over records
awk '(NR%2==0) {print}'
```

The third program is an awk program that receives output from the second program. It only prints records that are divisible by 2. This is done, to remove td tags that don't contain the information we need. The output of this program is piped to the fourth program.

Listing 6: Fourth program

```
#this awk program is responsible for concatenation of lines
awk '{line=line "," $0}  (NR%4==0) {print substr(line,2); line=""}'
```

The main function of the fourth program is to concatenate the records of the output it receives from the third program. The program is an awk program which initialises a variable called line and adds the current record to the variable. The lines are separated using a comma. The program concatenates the first four records then re-initialises the line variable to null and the loop begins again until end of file is reached. The output of this program is piped to the fifth program.

Listing 7: Fifth program

```
#sed program adds the single header line
sed -e '1 i\Timestamp,Latitude,Longitude,MinSeaLevelPressure,MaxIntensity'
    > "$2"
```

The last program is a sed program. It takes the output of the fourth program and inserts the text provided at the first line. The result of this is written to the file which is passed as the second argument of the bash script.

3 Storm Plots

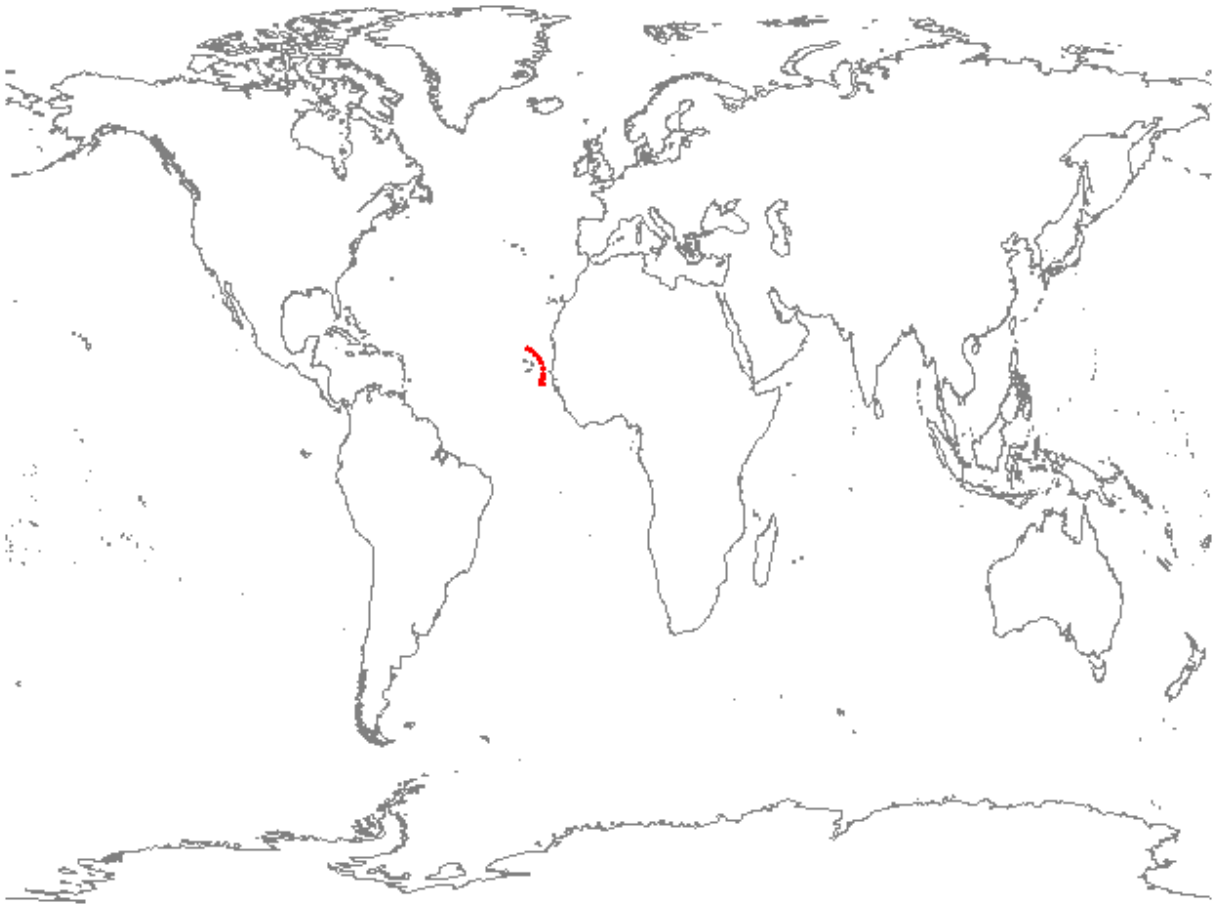

Figure 1: Storm plot diagram produced from the al102020.kml file

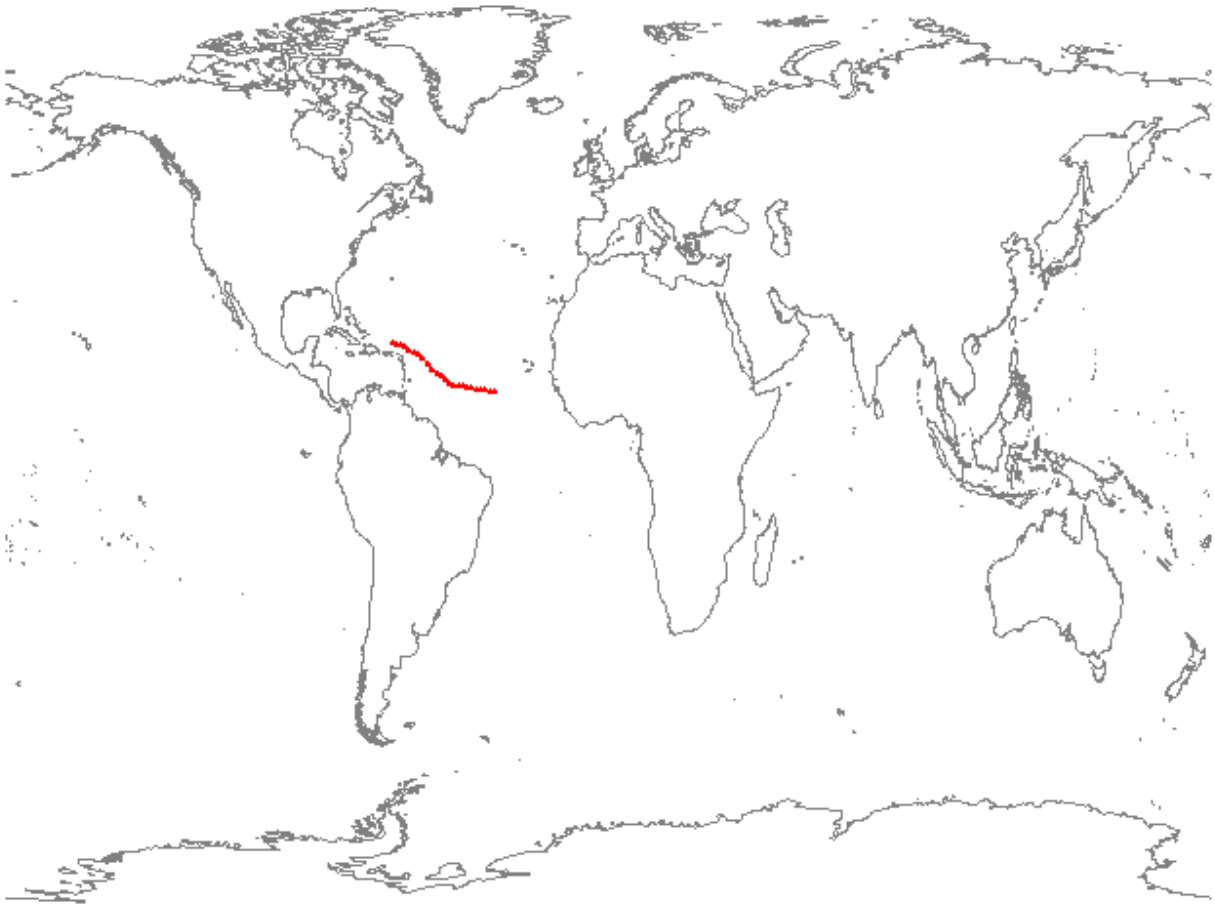

Figure 2: Storm plot diagram produced from the al112020.kml file

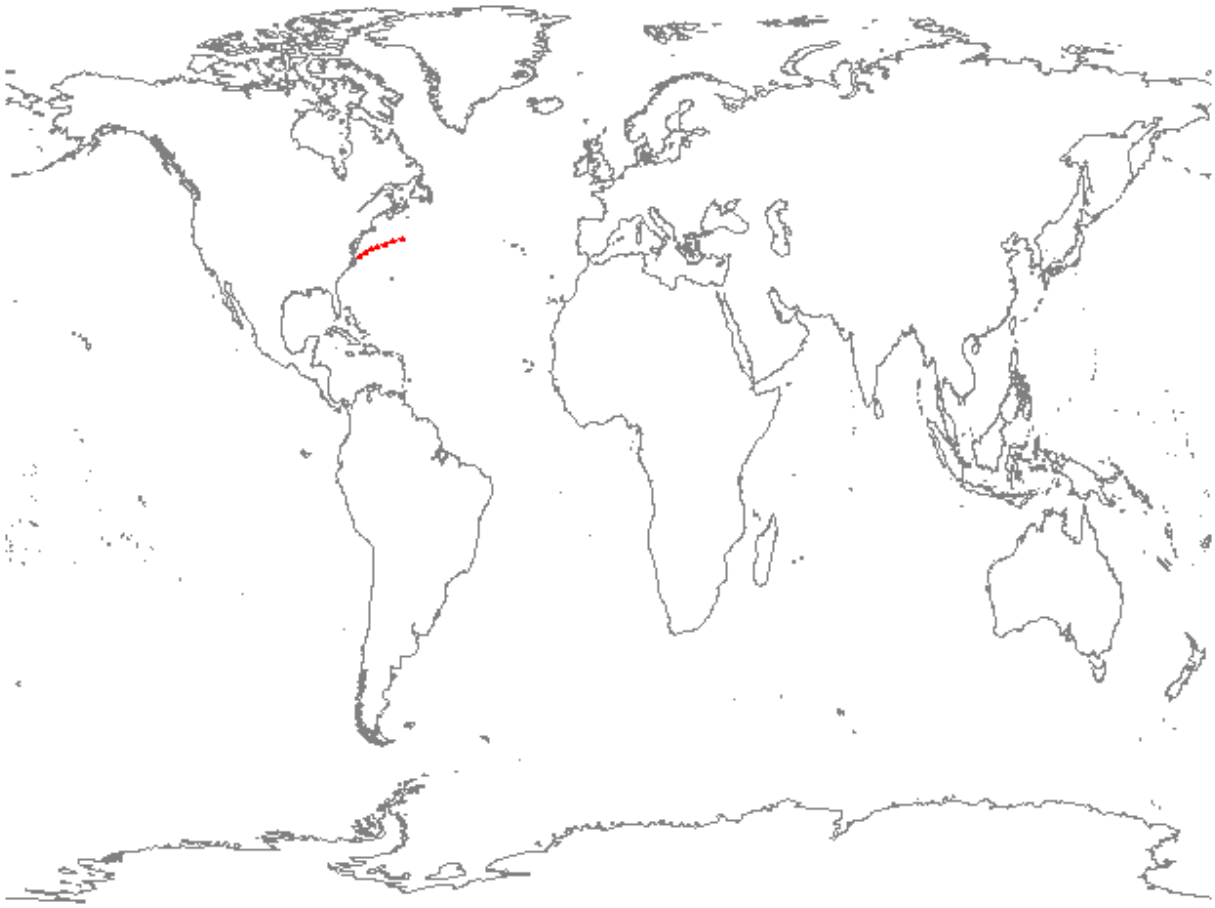

Figure 3: Storm plot diagram produced from the al122020.kml file